Computational Numerical Statistics
Assignment 4

Migla Miskinyte Reis

29.10.20

Exercise 4. (0.1 points) **Points:** _______

Consider

$$I = \int_0^1 e^{-x}(1-x)^2 \, dx$$

Use m= 10000 and set.seed(1234) in all experiments.

4.1 Use the R function integrate() to compute the value of the integral

```
f<-function(x){exp(-x)*(1-x)^2}

f_int<-integrate(f,lower=0,upper=1);f_int
```

```
## 0.2642411 with absolute error < 2.9e-15
```

4.2 Now, we will estimate I using the antithetic-variable-based Monte Carlo method and report an estimate of variance I_{ant} using this method.

```
set.seed(1234)
m=10000
g <- function(x){exp(-x)*(1-x)^2}
x=runif(m/2,0,1)
I.hat1=mean(g(x))
I.hat2=mean(g(1-x))
I.hat.a =(I.hat1+I.hat2)/2; I.hat.a
```

```
## [1] 0.263984
```

```
V.a <- 1/m*(1+cor(g(x),g(1-x)))*var(g(x)); V.a
```

```
## [1] 2.130907e-06
```

4.3 Next we will use MC method to estimate I and report I_{MC}.

```r
set.seed(1234)
m=10000
x=runif(m,0,1)
g <- function(x){exp(-x)*(1-x)^2}
I.hat=(1-0)*mean(g(x));I.hat
```

```
## [1] 0.2626583
```

```r
V.I.hat = var(g(x))/m*(1-0)^2;V.I.hat
```

```
## [1] 7.767264e-06
```

4.4 Next, we will estimate percentage of variance reduction when using I_{ant} instead of I_{MC} using the equation:

$$percentage = 100(1 - \frac{V(I_2)}{V(I_1)})$$

```r
100*(1-V.a/V.I.hat)
```

```
## [1] 72.56554
```

4.5 Next, we will use above described variance reduction methods to estimate I_{ant} instead of I_{MC} using differing m (100, 500, 1000, 5000, 1000). First, we create a function for antithetic-variable-based methodand estimate I and I_{ant} for different m.

```r
function_ant<-function(m){
  set.seed(1234)
  f <- function(x){exp(-x)*(1-x)^2}
  x=runif(m/2,0,1)
  I.hat1=mean(g(x))
  I.hat2=mean(g(1-x))
  I.hat.a =(I.hat1+I.hat2)/2; I.hat.a
  V.a <- 1/m*(1+cor(g(x),g(1-x)))*var(g(x)); V.a
  return(c(I.hat.a,V.a))
}

m100<-function_ant(100)
m500<-function_ant(500)
m1000<-function_ant(1000)
m5000<-function_ant(5000)
m10000<-function_ant(10000)
```

Next, we create a function for usual Monter Carlo variance reduction method and estimate I and I_{MC} for different m.

```r
function_mc<-function(m){
  set.seed(1234)
  x=runif(m,0,1)
  g <- function(x){exp(-x)*(1-x)^2}
```

```r
  I.hat=(1-0)*mean(g(x));I.hat
  V.I.hat = var(g(x))/m*(1-0)^2;V.I.hat
  return(c(I.hat.a,V.I.hat))
}

m100m<-function_mc(100)
m500m<-function_mc(500)
m1000m<-function_mc(1000)
m5000m<-function_mc(5000)
m10000m<-function_mc(10000)
```

We also need to estimate percentage for variance reduction for each m between our methods.

```r
perc100<-100*(1-m100[2]/m100m[2])
perc500<-100*(1-m500[2]/m500m[2])
perc1000<-100*(1-m1000[2]/m1000m[2])
perc5000<-100*(1-m5000[2]/m5000m[2])
perc10000<-100*(1-m10000[2]/m10000m[2])
```

Finally, we will make a table with our observations. We can see that variance using I_{MC} is lower than I_{ant} and that overall variance decreases when we increase m.

```r
mm<-cbind(m100m,m500m,m1000m,m5000m,m10000m) #created a matrix of vectors for table
row.names(mm)<-c("I_ant Estimate","I_ant Variance")
ma<-cbind(m100,m500,m1000,m5000,m10000)
row.names(ma)<-c("I_MC Estimate","I_MC Variance")
perc<-matrix(c(perc100,perc500,perc1000,perc5000,perc10000),1,5)
row.names(perc)<-c("% var reduction")
table<-rbind(ma,mm,perc)
colnames(table)<-c("100", "500", "1000", "5000", "10000")
knitr::kable(table)
```

	100	500	1000	5000	10000
I_MC Estimate	0.2444931	0.2593640	0.2599140	0.2624219	0.2639840
I_MC Variance	0.0001676	0.0000398	0.0000204	0.0000042	0.0000021
I_ant Estimate	0.2639840	0.2639840	0.2639840	0.2639840	0.2639840
I_ant Variance	0.0008308	0.0001454	0.0000779	0.0000156	0.0000078
% var reduction	79.8209997	72.6467373	73.7666389	73.3153215	72.5655438